Few-Shot Learning with Embedded Class Models and Shot-Free Meta Training

Anonymous ICCV submission

Paper ID 3571

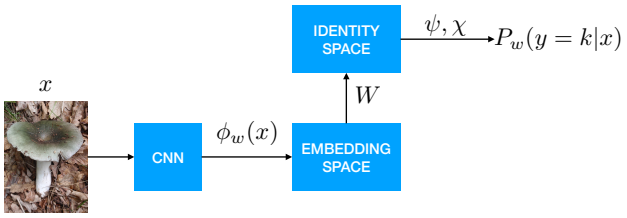

Figure 1. Shown in the figure is an overview of our method. The block diagram shows the various concepts we have introduced in the paper and the stage in the pipeline in which they are used. A key feature of our approach is that the embedding space and the identity space need not be the same. The membership to an identity is calculated in the identity space endowed with the metric χ and identity function ψ.

1. Overview of our approach

Figure 1 shows the block diagram of our approach and the different components of our approach such as ϕ_w, ψ, χ

2. Additional Experimental Results

In this section, we validate our implementation of Prototypical Networks and show additional comparisons. We also present additional results of our method compared to the state-of-the-art. We finally show the effect of choices during training on performance.

2.1. Comparison to Prototypical Networks

Validation of Implementation We first validate our implementation of Prototypical Networks [10]. Shown in Table 1 is the comparison of our implementation of [10] to the results reported in [10]. From this table, we see that the results are statistically the same. This shows that our implementation is correct and is able to reproduce results reported in [10].

Additional Comparisons We next show results of our method on additional training scenarios as in Prototypical Network [10]. In addition to training with 1-shot 5-way and 5-shot 5-way, experiments were also shown using a 1-shot 30-way and a 5-shot 20-way training setup. We present this result in Table 2 on three datasets namely miniImagenet, tieredImagenet and CIFAR Few-Shot. These results use the C64 network architecture. This table shows that across datasets, our method shows a significant improvement for the 1-shot 5-way scenario. For the 5-shot 5-way scenario we see improvements but they are not as significant as compared to the 1-shot 5-way scenario. The goal of this table was to show that our improvements are not restricted only to the 5 way scenario.

2.2. Comparison to the State-of-the-art

We show additional comparisons to the state-of-the-art for miniImagenet in Table 3. Here, in addition to the results shown in the paper, we show results for the ResNet-12 Variant with the identity living in $2\times$ the dimension of the embedding. From this table, we see that increasing the dimension of the identity helps with the 1-shot case, but we see a slight drop in accuracy for the 5-shot case.

However, we see that for the tieredImagenet dataset, increasing the dimension of the identity helps in all scenarios. Shown in Table 4, are two additional results, namely the effect of increasing the dimension of identities for the ResNet-12 network and its variant. We see that for this dataset, our method is able to outperform the state-of the-art for both 1-shot and 5-shot.

2.3. Effect of Choices in Training

We had outlined in the manuscript that comparing to the state-of-the-art is extremely hard as there are numerous choices that different algorithms make. Here we outline a few choices and show their effect on the performance.

Effect of Optimization algorithm In the original implementation of Prototypical Networks [10], ADAM [4] was used as the optimization algorithm. However, most newer algorithms such as [6, 3] use SGD as their optimization algorithm. This result of using different optimization algorithms is shown in Table 5. Here, we show the performance of our algorithm on the miniImagenet dataset using a ResNet-12 model. From this table we see that, while for

Testing Scenario	Training Scenario	As reported in [10]	Our implementation of [10]
1-shot 5-way	1-shot 30-way	49.42 ± 0.78	48.36 ± 0.42
5-shot 5-way	5-shot 5-way	65.77 ± 0.70	65.49 ± 0.35
5-shot 5-way	5-shot 20-way	68.30 ± 0.66	68.29 ± 0.34

Table 1. Comparison of results from [10] to that our implementation of Prototypical Network [10] using the C64 network architecture. The table shows the accuracy and 95% percentile confidence interval of our method averaged over 2,000 episodes on the miniImagenet dataset.

Dataset	Testing Scenario	Training Scenario	Our implementation of [10]	Our Method
miniImagenet	1-shot 5-way	1-shot 30-way	48.36 ± 0.42	**52.35 ± 0.40**
miniImagenet	5-shot 5-way	5-shot 20-way	68.29 ± 0.34	**69.31 ± 0.33**
tieredImagenet	1-shot 5-way	1-shot 30-way	47.43 ± 0.45	**53.90 ± 0.44**
tieredImagenet	5-shot 5-way	5-shot 20-way	69.47 ± 0.38	**71.15 ± 0.38**
CIFAR Few-Shot	1-shot 5-way	1-shot 30-way	57.03 ± 0.50	**60.68 ± 0.49**
CIFAR Few-Shot	5-shot 5-way	5-shot 20-way	75.51 ± 0.37	**75.69 ± 0.37**

Table 2. Comparison of results from our method to our implementation of Prototypical Network [10] using the C64 network architecture. The table shows the accuracy and 95% percentile confidence interval of our method averaged over 2,000 episodes on different datasets.

Algorithm	1-shot 5-way	5-Shot 5-way	10-shot 5-way
Meta LSTM [7]	43.44	60.60	-
Matching networks [13]	44.20	57.0	-
MAML [2]	48.70	63.1	-
Prototypical Networks [10]	49.40	68.2	-
Relation Net [12]	50.40	65.3	-
R2D2 [1]	51.20	68.2	-
SNAIL [5]	55.70	68.9	-
Gidaris*et al*. [3]	55.95	73.00	-
TADAM [6]	58.50	76.7	80.8
MTFL [11]	61.2	75.5	-
LEO [9]	61.76	77.59	-
Our Method (ResNet-12)	59.00	77.46	82.33
Our Method (ResNet-12) 2x dims.	60.64	77.02	-
Our Method (ResNet-12) Variant	59.04	**77.64**	**82.48**
Our Method (ResNet-12) Variant 2x dims	60.71	77.26	-

Table 3. Performance of 4 variants of our method on miniImagenet compared to the state-of-the-art. The table shows the accuracy averaged over 2,000 episodes.

the 1-shot 5-way the results are better with ADAM as opposed to SGD, we see that the same does not hold for the 5-shot 5-way and 10-shot 5-way scenarios. This shows that SGD generalizes better for our algorithm as compared to ADAM.

Effect of Number of and tasks per iteration. TADAM [6] and Gidaris *et al*. [3] use multiple epsiodes per iteration. They refer to this as tasks in TADAM [6], which uses 2 tasks for 5-shot, 1 task for 10-shot and 5 task for 1-shot. We did not perform any such tuning and instead defaulted it to 8 episodes per iteration based on Gidaris *et al*. [3]. We also experimented with 16 episodes per iteration. However, this led to a loss in performance across all testing scenarios. Table 6, shows the performance numbers on miniImagenet dataset using the ResNet-12 architecture and trained using ADAM [4] as the optimization algorithm. From this table we see that for all the scenarios 8 episodes per iteration has a better performance.

Algorithm	1-shot 5-way	5-Shot 5-way	10-shot 5-way
MAML [2]	51.67	70.30	-
Prototypical Networks [8]	53.31	72.69	-
Relation Net [12]	54.48	71.32	-
LEO [9]	65.71	81.31	-
Our Method (ResNet-12)	63.99	81.97	85.89
Our Method (ResNet-12) 2x dims.	**66.87**	**82.64**	-
Our Method (ResNet-12) Variant	63.52	82.59	**86.62**
Our Method (ResNet-12) Variant 2x dims	**66.87**	82.43	-

Table 4. Performance of our method on tieredImagenet as compared to the state-of-the-art. The table shows the accuracy averaged over 2,000 episodes.

Choice	1-shot 5-way	5-Shot 5-way	10-shot 5-way
Optimization Algorithm (ADAM)	**59.41**	76.75	81.33
Optimization Algorithm (SGD)	59.00	**77.46**	**82.33**

Table 5. Performance of our method on miniImagenet using a ResNet-12 model with different choices of optimization algorithm. The table shows the accuracy averaged over 2,000 episodes.

Choice	1-shot 5-way	5-Shot 5-way	10-shot 5-way
8 episodes per iteration	**59.41**	**76.75**	**81.33**
16 episodes per iteration	58.22	74.53	78.61

Table 6. Performance of our method on miniImagenet using a ResNet-12 model with different choices of optimization algorithm. The table shows the accuracy averaged over 2,000 episodes.

References

[1] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *CoRR*, abs/1805.08136, 2018. 2

[2] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2

[3] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 1, 2

[4] D. P. Kingma and J. L. Ba. ADAM: A method for stochastic optimization. *International Conference on Learning Representations 2015*, 2015. 1, 2

[5] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018. 2

[6] B. N. Oreshkin, P. Rodríguez, and A. Lacoste. Improved few-shot learning with task conditioning and metric scaling. In *NIPS*, 2018. 1, 2

[7] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2

[8] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-learning for semi-supervised few-shot classification. *CoRR*, abs/1803.00676, 2018. 2

[9] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. *CoRR*, abs/1807.05960, 2018. 2

[10] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4080–4090, 2017. 1, 2

[11] Q. Sun, Y. Liu, T. Chua, and B. Schiele. Meta-transfer learning for few-shot learning. *CoRR*, abs/1812.02391, 2018. 2

[12] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[13] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NIPS*, 2016. 2